LLNL-PROC-656456

LAWRENCE LIVERMORE NATIONAL LABORATORY

A Mesh-free DSD Front Tracker for an Arbitrary HE Boundary

J. Yao

July 3, 2014

The 15th International Detonation Symposium
San Francisco, CA, United States
July 7, 2014 through July 11, 2014

Disclaimer

A Mesh-free DSD Front Tracker for an Arbitrary HE Boundary

Jin Yao

WCI-B Division

Lawrence Livermore National Laboratory, Livermore, California, U.S.A

Abstract. A new mesh-free DSD front tracker that explicitly solves the detonation front propagation is implemented for an arbitrarily specified HE boundary. Compared to previously existing mesh-based DSD implementations, the new method is advantages with reduced cost, improved accuracy, minimized data space, and enhanced capabilities. The new DSD front tracker puts marked particles on the HE front and tracks the motion of the particles in the front normal direction, governed by a time dependent ODE system. The difficulty associated with an implicit DSD boundary treatment is much reduced by an explicit methodology for arbitrarily specified HE boundary geometry. The issue associated with the change of front topology is resolved with an efficient local inclusion test. Particles are redistributed on the front to maintain the desired resolution. This DSD front tracker has been validated with comparisons to exact and reactive flow solutions of rate-stick problems.

Introduction

Programmed HE burn with a detonation shock dynamics (DSD) model is an efficient method for the lighting time calculation compared to reactive flow modeling which requires high resolution for the reaction zone. Existing DSD implementations solve for the level-set PDE (or some equivalent PDE) with a mesh and apply an angle condition on the HE/inert boundary. Among the existing DSD algorithms, a finite-difference method can be easily performed on a Cartesian mesh. However, the boundary treatment becomes complicated because both the detonation front and the HE boundary are implicitly defined [1],[2]. With a renormalization of the distance field, a narrowband methodology can reduce the cost, but would not converge to the true solution with even a relatively trivial concave geometry [3]. Another method using a least-squared fitting of the lighting front utilizing an existing hydro mesh is also implemented in a narrowband fashion but the boundary condition is still difficult to apply even when the HE boundary is explicitly specified with a body-fitting mesh [4]. Because mesh resolution cannot be adjusted, local high curvature regions cannot be resolved with a mesh-based DSD method. In addition, a mesh-based method requires relatively big storage space.

To reduce the above difficulties, a DSD algorithm with both the detonation front and the HE boundary explicitly expressed is desirable. Such a method may provide an easier boundary treatment, an adjustable local resolution for a more accurate and efficient solution, with a smaller data space.

Recently, we have implemented a mesh-free DSD front tracker that admits an arbitrarily specified HE boundary. The new algorithm tracks particles distributed on the lighting front according to a given DSD curvature law, with a curved fitting of neighbor particle positions. The DSD boundary angle condition is used to explicitly derive the position of a boundary particle. The challenge with topology change is treated by employing an efficient geometrical inclusion method, and utilizing a uniform virtual background Cartesian grid. The necessity with redistribution of particles is performed by a relaxation method based on distance (or curvature) for maintaining accuracy.

With this new algorithm there is no need to solve the governing nonlinear PDE as with the previous implemented DSD methods. Instead, a set of time-dependent ODEs derived from the DSD evolution equation [5] is used to advance a detonation front. This PDE to ODE conversion is third order accurate in space and introduces no error in time.

In this article, we will describe the mathematical aspects and numerical techniques for front motion, topology change, boundary treatment, and surface management associated with this mesh-less DSD front tracker. Some 2D numerical examples that verify and validate the new method then follow.

HE Boundary Presentation: with Fixed Control Points that Define Boundary Faces

In practice, boundary geometry usually can be specified with a set of control points and a rule to connect them to represent boundary faces. For example, the rate-stick geometry can be defined with four points. To define a boundary in two-dimensions, the control points can be ordered counter-clock-wise with two consecutive points defining a face. A face can be planar or curved depending on if continuity of slope at a control point is in consideration. In an axi-symmetrical geometry, the "faces" are the faces in two-dimensions revolved around the axis. Rather complicated boundary geometry can possibly be defined with a small set of control points (this can be seen with numerical examples later in this article). The smallness of boundary data is usually also true in three-dimensions.

Burn Front Representation: with Particles and a (Distance-based) Neighbor Relationship

A detonation front at a time step is presented by particles distributed on the front (fig. 1). To completely describe a given detonation front, it is sufficient to assign for each particle a set of neighbor particles. The local geometry of a front is described by the positions of neighbor particles.

We select the neighbors with distances between particles by specifying a search-length $\boldsymbol{s}$ which can be the desired resolution in the problem. At any time step, a particle looks for its neighbors within the distance $\boldsymbol{s}$. At the next time-step, some neighbor particles may move into or out of the neighborhood of a given particle. This change can be traced locally by the neighbor relationship.

Each particle is assigned a unit normal vector as their direction of motion. A natural treatment is to fit a plane (in 3D) or a line (in 2D) with particles in the neighborhood and so this is our choice. The direction of the normal vector is determined by specifying the orientation of the particles in the neighborhood. In 2D a particle carries a left and a right neighbor to define the orientation. In 3D, the neighbors are ordered counter-clock-wise in the fitting plane with particles projected on the plane.

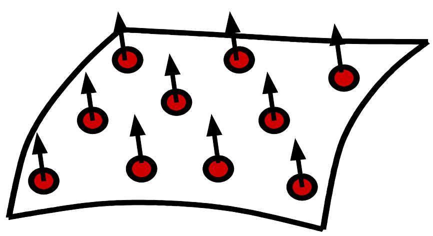

Fig.1. Particle presentation of a local detonation front in 3D is shown. Vectors are in the normal direction of propagation of the detonation.

Front Motion: with a PDE to ODE Conversion of the Equation of Motion of a Detonation

A given particle **P** on a detonation front move in the normal direction with a velocity D_n computed from a given D_n-κ relation (subscript n for normal). The front curvature κ is obtained from fitting a curved surface to neighbor particles in the surface normal coordinate defined for **P** (fig. 2).

To the leading order approximation of the DSD theory, D_n, the detonation front velocity in the normal direction, is a function of front curvature κ, i.e. $D_n = D_n(\kappa)$. The motion of the front is described by a partial differential equation

$$\frac{\partial \varphi}{\partial t} + D_n(\kappa)\,|\nabla \varphi| = 0. \qquad (1)$$

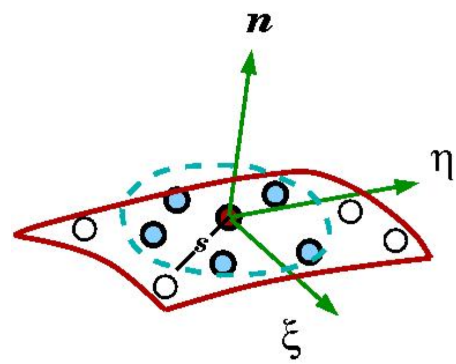

*Fig.2. The local front normal coordinate for a given particle **P**. The front geometry is defined with a quadratic fitting with neighbor particles (filled circles within distance **s**) of **P**. Unfilled circles mark particles that are not neighbors of **P**.*

Here the surface $\varphi(\vec{r},t) = 0$ defines the position of the front in space and time. We do not solve the level-sets; merely we track the position of the front.

The normal vector at a particle **P** on a given front $\varphi(\vec{r},t) = 0$ is defined by $\hat{n} = \vec{\nabla}\varphi / |\vec{\nabla}\varphi|$. In three-dimensions, a local Cartesian-coordinate system $(\hat{n}, \hat{\xi}, \hat{\eta})$ can be defined by setting the origin at **P** with the normal at it being the n-axis; ξ-axis and η-axis are the principal tangential unit vectors. In this intrinsic normal coordinate, the front geometry can be generally expressed by n = $h(\xi, \eta)$ in a neighborhood of **P**. We assume here the surface that represents a detonation front is sufficiently smooth to well define curvature. Such a surface may take the form of a Taylor expansion with a third order spatial accuracy in the format

$$n = \frac{a\xi^2}{2} + b\xi\eta + \frac{c\eta^2}{2} + d\xi + e\eta + f. \qquad (2)$$

Where the coefficients *a, b, c, e, f,* and *g* are considered functions of time that carry the change of geometry of the detonation front.

A reduced form with the front expansion eq. (2) in the surface normal coordinate can be derived. Let us consider the moment $t = T_0$, without loss of generality T_0 can be set to 0. Because the front passes the origin, one must have f $(0) = 0$.

We compute the normal vector at the origin with our choice of $\varphi = n - h(\xi, \eta)$ using eq. (2). This normal by definition is proportional to the vector $\vec{\nabla}\varphi = \hat{n} - (a\xi + b\eta + d)\hat{e}_\xi - (b\xi + c\eta + e)\hat{e}_\eta$. At the origin $(0, 0, 0)$, the normal must only have the n-component therefore, therefore we also have $d(0) = 0$ and $e(0) = 0$. We set $b(0) = 0$ for $(\hat{e}_\xi, \hat{e}_\eta)$ are the directions with the principal curvatures. The local geometry has symmetry about the *n*-axis.

To track the motion of the particle initially at the origin, we substitute $\varphi = n - h(\xi, \eta)$ into the equation of motion eq. (1), then take arbitrarily small ξ and η, and expand the equation to third order, with some algebraic manipulations. The equation of motion is reduced to a set of ODEs that $\dot{a} = a^2 D_n(\kappa), \dot{c} = c^2 D_n(\kappa), \dot{f} = D_n(\kappa)$.

The motion of a particle **P** is governed by the above ODEs. The local front geometry will be symmetric above *n*-axis to third order without introducing error in time. For each marked particle **P** on the detonation front, a corresponding local surface normal coordinate can be defined as above.

The curvature κ at a particle is then $-(a + c)$. The quadratic front fitting to the positions of neighbor particles is performed at the start of each time step. An ODE integrator can be used to update the position of a particle at the end of this time-step with a high accuracy in time.

Particles maybe added / deleted / rearranged on the front for an ideal spatial distribution for the next step. The procedure described is iterated until the entire HE region is lit.

Linear Stability and the Size of Time-steps

For a near CJ diverging detonation wave, the curvature is positive and the detonation speed is below its CJ value. In this case, a linear stability analysis of the ODE system shows negative growth factors. The time integration is then stable. For a converging detonation, we take the CJ velocity to propagate the detonation and there is no uncontrolled growth introduced with this choice. Rearranging particles tends to smooth the front and this would also help to stabilize this numerical scheme. However, the time step should be limited in order to avoid unnecessary reordering of neighbor particles. For the purpose of testing we have taken constant time steps.

Treatment of Topology Change: with an Inclusion Test Utilizing a Cartesian Virtual Grid

The information about if a marked particle has moved out of the boundary, or into burned region has to be provided by an inclusion test. Directly applying conventional inclusion tests for the case of many points is inefficient. We have implemented an inclusion test based on locality to improve the efficiency.

We use a special length l_c as the unit of length to scale the problem. After dividing coordinates by l_c, each particle is contained in a unit cube (i, i+1; j, j+1; k, k+1) with (i, j, k) being the integer floor values of the coordinates of the particle position.

Those virtual-cubes that contain the HE boundary are identified by finding the intersections of the unit cubes and the boundary (fig. 3). We have implemented a directional-walking intersection algorithm [6] for boundaries consists of planar or Hermite cubic spline faces. We perform a local quadratic fit to a detonation front to compute its intersection with unit cubes. When a particle is located in a cube that is sliced by either the HE boundary or the detonation front, we check its relative position to the boundary (or front) with a local inclusion test inside this unit cube only. This approach makes the inclusion test very efficient.

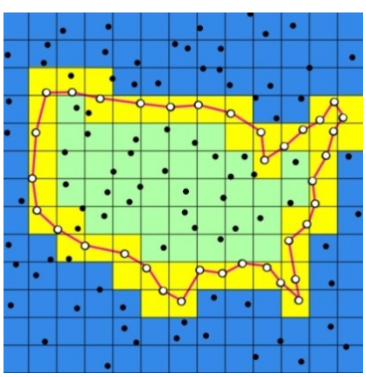

Fig. 3. Utilizing background virtual cubes, only the particles contained in the boundary cubes need to be test for inclusion with the boundary faces crossing a cube that contains a given particle. Each virtual cube is a unit one after scaling by l_c.

The size of a virtual cube l_c should be a fraction of the volume of the HE region divided by its surface area. Then only those particles contained in the boundary cubes (the cubes crossed by boundary) need to be tested for inclusion with only the boundary faces that cross a given cube. These virtual cubes maybe used later for a linear neighbor search for an interpolation of lighting times carried by particles that define the detonation fronts (lighting time contours). In general l_c is much bigger than the mesh resolution required. Because only those virtual cubes crossed by boundary and front need to be stored at a time, the data storage for the virtual cubes is very small.

Boundary Treatment: Explicitly Apply the DSD Boundary Angle Condition

A detonation front is presented by particles distributed on the front. The front curvature is obtained from fitting a curved surface to neighbor particles. For a boundary particle, this fitting is constrained by a material specific angle between the detonation front and the HE boundary. For a boundary particle in two-dimensions, its two nearest inner neighbor particles are employed to fit a circle that intersects the boundary with the angle ω_c. The boundary particle is put at the intersection (fig. 4). In three-dimensions, the idea is similar: fit for each boundary particle a quadratic surface to the updated positions of its interior neighbors, with an angle constraint applied at the intersection of the front and the boundary. The associated algebra is not hard to perform for planar boundary geometry. For a curved boundary the solution is a bit harder to handle, but a local planar fitting of the boundary can be done for an approximate solution. A finer solution would require iterations.

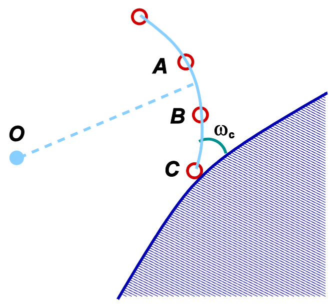

Fig. 4. A circle that passes interior neighbors A, B and intersects the boundary with the DSD angle ω_c is constructed to locate boundary particle C.

In three-dimensions, the boundary condition is to be applied in a plane that is perpendicular to the boundary. The intersection of the quadratic fitting of the front and this plane needs to be computed for applying the DSD boundary angle in this plane. Nevertheless, it becomes a two-dimension problem.

Front Management: with Adding, Deleting and Relaxing/Rearranging the Particles

As a principle we drop particles that move out of boundary or into burned regions. Currently we choose to control the resolution with a length α as the desired size of a face. In two-dimensions this can be simply done by keeping the number of particles on a piece of front of an arc-length L being the integer portion of L divided by α, and then evenly place the particles. The criterion for placing the particles can also be performed by a relaxation rule based on curvature such as $\kappa\alpha$ = *const* where α is the dimension of a face of the front. A curvature based distribution of particles allows a variable resolution and improves accuracy for regions with higher curvature. Relaxation based on distance or curvature can be done in 3D.

Local insertion of particles for maintaining the distance between nearest particles can *always* be performed for a given particle and its neighbors to control the front resolution. The associated search of nearest particles with a search-length is local with the known neighbor information.

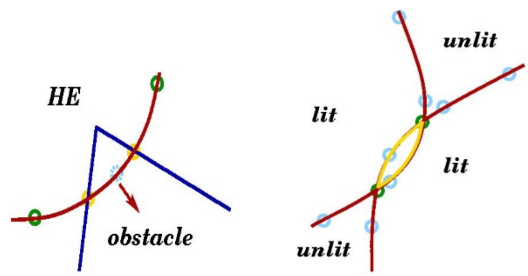

Fig. 5. Particles out of boundary or into lit regions (identified with the inclusion test) after a time advancement, are to be dropped.

Neighbor Search and Change of Topology

As described previously, a given particle carries a set of neighbor particles and the search of neighbors is based on a search length that reflects the local resolution desired. When multiple detonations impact, because only a first order accuracy is expected where some particle(s) run into regions that are burned, there is no need to calculate the intersection of two fronts. One merely keeps using a quadratic fit to the updated neighbors for front geometry for topology change.

A Small Data Space: with a Reduced Complexity

In practice, only an ignorable storage is needed for specifying the boundary geometry. Furthermore, at any time step the information needed to advance the front is carried by only the positions of particles on the current front. A front processed by the new DSD tracker is a (k–1)-dimensional profile embedded in k-dimensional space (k = 2, 3). The spatial complexity for a mesh-based DSD method is reduced by a dimension. The proposed method has third-order accuracy in space, thus a high resolution is not required unless a local high front curvature needs to be resolved. For a given accuracy, an optimized number of particles can be distributed to describe the front. Those virtual-cubes that are crossed by the boundary and the front at a time step need to be carried. However, the corresponding storage is also rather small.

Since the spatial complexity is reduced with our mesh-free methodology, a minimal data space is required (to store only the boundary geometry, the current particle positions, and the virtual cubes) to advance the detonation front. Effort toward parallelization is possibly unnecessary for even a rather big-sized DSD problem. This might be one of the major advantages associated with a mesh-less DSD implementation. Because the proposed scheme depends solely on neighbor relationship, a parallelization is not hard to perform if necessary.

Supply Lighting Times: with Local Interpolations

The marked particles distributed on each of the fronts (time contours), provides a table of lighting times. In a hydro simulation, a lighting time is assigned at each HE node (or zone-center) from the data carried by the marked particles. This is achieved by utilizing the virtual cubes for the inclusion test. For a virtual cube that contains a given node **N**, one collects a list of marked particles that are also contained in this cube (fig. 6). The detonation arrival time and particle coordinate for each particle in the list are used to construct a local lighting-time field with a quadratic least squared fit. Then the lighting time at the node **N** is obtained. The neighbor search with this operation costs linearly for each virtual cube carrying a list of positions and lighting-times at particles. An example with this interpolation for a PBX-9502 HE lighting problem is shown in fig 7.

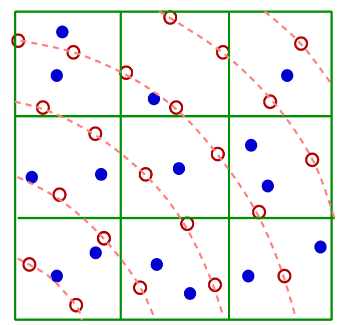

Fig.6. The lighting time at a solid dot is the interpolation of the values at the particles (circles) on time contours (dashed curves). Virtual cubes for inclusion also help the neighbor search.

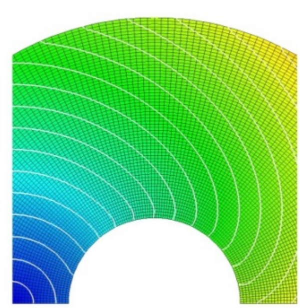

Fig. 7. The moving particle solution for a PBX-9502 circular channel detonation is converted into lighting times on a mesh with local interpolation. The DSD boundary angle is taken to $\pi/3$.

Numerical Examples: with Arbitrary Boundaries

Example A: with a Curved Geometry

As show in (fig. 8) the boundary of the HE region has a shape of a star-fish. With a mesh based DSD calculation, there would be many nodes to present the boundary. Here we only employed *ten* control points. A boundary face is a piece of a Hermite cubic-spline curve that ensures smoothness of the boundary (this is a common practice).

The D_n-κ relation is taken to $D_n = 1 - 0.2\kappa$, and the DSD boundary angle is set to $\pi/3$. The lighting starts with *30* particles evenly put on a flat detonation front. The front successfully turns the corners and expands into the center. Particles are redistributed for keeping the initial resolution. When the front hits the boundary again, some particles become exterior and are dropped. This procedure repeats itself until all the fronts become smaller than a preset threshold.

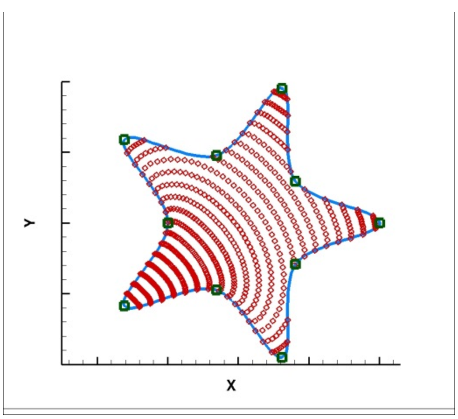

Fig. 8. Detonation fronts (lighting time contours) obtained with the particle DSD front tracker. With $D_n = 1 - 0.2\kappa$*, and a boundary angle* $\omega_c = \pi/3$*.*

Example B: with a Zigzag HE Channel

The HE region has a boundary consists of planar faces in (fig. 9). Twelve control-points are used to define this zigzag geometry. The same DSD parameters used in the last example are employed with a flat initial detonation front starts from the foot. The front successfully made turns around corners and kept intersecting the boundary at a *60* degree angle. This geometry is somehow more difficult to apply the boundary condition, for the boundary is not everywhere differentiable.

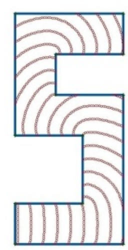

Fig. 9. The propagation of detonation in a zigzag channel is simulated with the mesh-less DSD front tracker, with the same DSD parameters.

Validation of the Mesh-less DSD Tracker

We have verified the scheme with rate-stick geometry first with the exact solution for a linear D_n-κ relation. Let the width of the stick be 2R, taking D_{CJ} as the velocity unit (then $D_n = 1 - \alpha\ \kappa$), R as the length unit to scale the system, and $\omega_c = \pi/4$, the system has a first integral in planar geometry

$$\frac{\alpha}{CT}\left(\arctan(\frac{1}{T}) + \arctan(\frac{C}{\sqrt{2}T}) - \frac{\pi T}{4}\right) = 1.$$

Where the term $T = (1 - C^2)^{1/2}$, with C the steady travelling velocity. For $\alpha = 0.1$, we get $C = 0.969723$, which is accurately recovered to 6 digits with this mesh-less detonation front tracker.

We also compared the numerical results obtained from this tracker with reactive flow simulation (CHEETAH) for rate stick problems with LX-17. We numerically derive the front velocity and curvature relation with CHEETAH data. Then we input the numerical D_n-κ relation and run the DSD front tracker, and obtained an almost exact match between the two. We consider it a justification to both the parameters we obtained from the LX-17 DSD study and this mesh-free front tracker.

The Numerical D_n-κ Relationship for LX-17

We employ a CHEETAH generated steady LX-17 detonation in rate-stick geometry represented by data-points on the front. We evenly divide the radius to intervals. At each joint of intervals, a search-distance D is used to look for data-points contained in the sphere centered at the joint with radius D. The selected data-points are used to fit a quadratic to obtain the curvature at each joint. The slope of the fitting curve provides information of normal front velocity at a joint. Then we obtain a (D_n, k) pair at each joint and a point on the D_n-κ curve for LX-17 rate-stick is obtained The small curvature extrapolation gives DCJ ~ 7.73 (cm/ms) which is in agreement with experimental results [9].

Comparison with CHEETAH Rate-stick Fronts

We have run our mesh-less DSD front checker with the numerical D_n-κ relation derived for LX-17, and a boundary angle of 0.4472π. The tracker generated a front visually overlaps the CHEETAH front. This shows that the DSD programmed-burn can adequately predict the detonation arriving time.

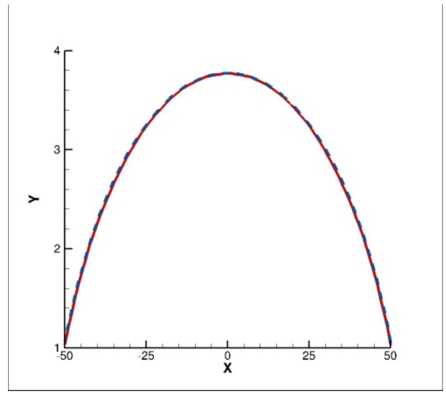

Fig.10. A steady detonation front in a 50mm rate-stick test generated with the mesh-less DSD tracker (dashed curve, using our numerical D_n-κ derived for LX-17) overlaps the front obtained from DNS with the CHEETAH model.

Conclusion

The new front tracker admits arbitrarily specified boundary geometry. Difficulties associated with a moving particle front tracker [7] are resolved with the new tracker. The particles are rearranged at times when their distribution becomes non-ideal. Particles can easily be added or removed from the front to maintain the quality of the front geometry. The issue about a particle moves out of valid region is addressed with an efficient inclusion test.

Besides being fast, the new DSD front tracker has the advantage of being compact. At any given time, only the boundary geometry and the lighting front geometry are needed to carry the calculation. This tracker is of high spatial accuracy (third order) thus a high resolution is not required unless a local high front curvature needs to be resolved (existing DSD methods has difficulties resolving high local curvature). The lighting time on a hydro mesh can be easily obtained with a local implementation of the particle data. The mesh-less tracker is validated by comparisons to detonation fronts generated with direct numerical simulation with a detailed reaction chemistry model (CHEETAH) for various DSD parameters. We conclude that this DSD tracker is capable to serve as an efficient, light weight tool for programmed burn with the detonation shock dynamics.

Future Work

With the capability to handle the high curvature near HE boundary, this mesh-less algorithm has the potential to simulate the dead-zone in corner-turning. We will spend effort with this possibility.

Acknowledgements

The author would like to express appreciation for the guide and encouragement from Dr. Larry Fried and Dr. Robert Rieben at LLNL, for the CHEETAH data from Dr. Peter Vitello at LLNL. The author also would like to thank Dr. Jim Reus and Dr. Frank Graziani at LLNL for reviewing the article.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

The paper has been reviewed and released for public audience with unlimited distribution. The LLNL review and release number is LLNL-PROC-656456.

References

[1]. T. Aslam, J. Bdzil, and D. S. Stewart, "Level-Set Techniques Applied to Modeling Detonation Shock Dynamics", *Journal of Computational Physics*, 126, 390-409, 1996.

[2]. J. B. Bdzil, R. J. Henninger, and J. W. Walter, "Test Problems for DSD2D", *LA-14277*, 2006.

[3]. M.A. Kenamond, "HE Burn Table Verification Problem", *LA-UR 11-03096*, 2011.

[4]. J. Yao, "A Geometrical DSD Lighting Algorithm on Arbitrary Unstructured Meshes", *UCRL-JRNL-213932*, Lawrence Livermore National Laboratory, 2005.

[5]. J. Yao, "A Time Dependent ODE Conversion of the Detonation Shock Dynamics Evolution Equation", *LLNL-TR-641948,* August 6, 2013.

[6]. J. Yao, "Directional Walking: A Simple Way to Intersect Arbitrary Geometries", *LLNL-JRNL-400055*, 2007

[7]. J.A. Sethian, "Level Set Methods and Fast Marching Methods", *Cambridge University Press*, 1999.

[8]. T. D. Aslam, J. B. Bdzil and L. G. Hill, "Extensions to DSD Theory: Analysis of PBX 9502 Rate Stick Data", *Eleventh International Detonation Symposium*, 1998.

[9]. Private communication with Larry Fried, Clark Souers, and Peter Vitello, *LLNL, 2013.*